DEFENSE TECHNICAL INFORMATION CENTER

Information for the Defense Community

DTIC® has determined on *04/02/2010* that this Technical Document has the Distribution Statement checked below. The current distribution for this document can be found in the DTIC® Technical Report Database.

☒ DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

☐ © COPYRIGHTED; U.S. Government or Federal Rights License. All other rights and uses except those permitted by copyright law are reserved by the copyright owner.

☐ DISTRIBUTION STATEMENT B. Distribution authorized to U.S. Government agencies only (fill in reason) (date of determination). Other requests for this document shall be referred to (insert controlling DoD office)

☐ DISTRIBUTION STATEMENT C. Distribution authorized to U.S. Government Agencies and their contractors (fill in reason) (date of determination). Other requests for this document shall be referred to (insert controlling DoD office)

☐ DISTRIBUTION STATEMENT D. Distribution authorized to the Department of Defense and U.S. DoD contractors only (fill in reason) (date of determination). Other requests shall be referred to (insert controlling DoD office).

☐ DISTRIBUTION STATEMENT E. Distribution authorized to DoD Components only (fill in reason) (date of determination). Other requests shall be referred to (insert controlling DoD office).

☐ DISTRIBUTION STATEMENT F. Further dissemination only as directed by (inserting controlling DoD office) (date of determination) or higher DoD authority.

Distribution Statement F is also used when a document does not contain a distribution statement and no distribution statement can be determined.

☐ DISTRIBUTION STATEMENT X. Distribution authorized to U.S. Government Agencies and private individuals or enterprises eligible to obtain export-controlled technical data in accordance with DoDD 5230.25; (date of determination). DoD Controlling Office is (insert controlling DoD office).

REPORT DOCUMENTATION PAGE

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
29-03-2010	Final Technical Report	01-10-2006 to 31-12-2009

4. TITLE AND SUBTITLE

Memory Reconsolidation and Computational Learning

5a. CONTRACT NUMBER

5b. GRANT NUMBER
N00014-07-1-0009

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Siegelmann, Hava T.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Massachusetts Amherst
Office of Grant and Contract Administration
70 Butterfield Terrace
Amherst, MA 01003-9333

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

ONR Reg Boston N62879
495 Summer Street, Room 627
Boston, MA 02210-2109

10. SPONSOR/MONITOR'S ACRONYM(S)
ONR

11. SPONSORING/MONITORING

12. DISTRIBUTION AVAILABILITY STATEMENT
Unrestricted

20100402019

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Memory models are central to Artificial Intelligence and Machine Learning, since memories hold knowledge and their updates are the heart of flexibility and adaptivity. Reconsolidation is a key process of human learning, modifying learned memories with new information. Reconsolidation has also been implicated in various disorders such as PTSD and OCD. Understanding the computational basis of reconsolidation is the focus of this work, as well as employing findings to create an improved memory methodology for a superior thinking machine. Through our research, we revealed basic principles of reconsolidation-like processes and included them in novel models. For the first time our neural memory models allow input dimension not to be constrained to a fixed size, similar to organic memory allocation for memories of greater importance or increased detail. The total number of memories is, in a practical sense, unbounded. Furthermore, beyond the state of the art, our memory system has the ability to process on-line as objects change. These attributes may be very beneficial in psychological modeling. Significantly, we were able to employ our models as powerful engineering tools by using them to recognize and cluster realistic images during change and movement, and to track in highly dynamic environments.

15. SUBJECT TERMS
memory reconsolidation, unbounded memories, clustering, tracking

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Siebelmann, Hava T.
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U	UU	20	19b. TELEPONE NUMBER (Include area code) (413) 577-4282

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at lest the year and be Year 2000 compliant, e.g., 30-06-1998; xx-08-1998; xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATES COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33615-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report, e.g. 1F665702D1257.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. AFOSR-82-1234.

5d. PROJECT NUMBER. Enter al project numbers as they appear in the report, e.g. 1F665702D1257; ILIR.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORS AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/ monitoring agency, if available, e.g. BRL-TR-829; -215.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.

Final Report 2010

Principal Investigator:

Hava T. Siegelmann, University of Massachusetts, Amherst, USA   hava@cs.umass.edu

Award: N000140710009

Title: Memory Reconsolidation and Computational Learning

Contents:

A. Scientific and Technical Objectives

B. Approach

C. Concise Accomplishments

D. Expanded Accomplishments

E. Productivity

F. Technology Transfer

A. Scientific and Technical Objectives

Reconsolidation is a storage process occurring during retrieval, in which an existing memory becomes labile and amenable to being updated. The process is implicated in learning and memory flexibility when healthy; it is correlated with amnesia and compulsive disorders when corrupted. The process of reconsolidation is observed both in neurophysiological and in psychological studies. The underlying objective of this project was to elucidate a functional and algorithmic understanding of reconsolidation in order to comprehend the particular benefits the process provides humans to better adapt in dynamic environments. The project also examined the disadvantages stemming from a lack of flexibility when reconsolidation does not work well. The intent is to employ an empirical understanding of the process and introduce a significantly improved thinking machine methodology. This new methodology can employ computational learning algorithms with the functional benefits resulting from reconsolidation. Applications include the design of machinery for recognizing dynamically changing concepts with contextual sensitivity, tracking movements in naturally changing environments, and clustering objects during monotonic changes. This research may start a new subfield of machine learning since current recognition and clustering applications rely on static objects and multiple repetitions of the sample set of images, while reality may provide dynamic data characterized by trajectories. Significantly, our new approach can successfully interface with this sort of realistic dynamical input. Furthermore, unlike computerized memories and other state of the art cognitive architectures, our memory system has the ability to process on-line and in real-time as objects change. Such a novel computational memory also has the potential to underlie improved methods of human-robot interaction in the future, relying on more human-like representations and functionality.

B. Approach

We allocated our efforts along the following activity tracks:
A: Relation to biology and psychology: A1. Analyzing existing biological and behavior data following reconsolidation: We asked, what do updated memories contain after the reconsolidation process integrates existing memories with newer experience? Are older memories gone after memory changes, as seen in hippocampus place cells CA3 and CA1 (Neuron 2005) and as seen in the psychophysics of morphing faces (Vision Res 2007)? And, why doesn't memory change when the series of inputs is not ordered monotonically? A2. Proposing a mathematical theory and finding principles that enable prediction and explicate memory attractor changes during reconsolidation. And investigating how changes are affected by the relative ordering of the input series.
B: Building memory software to test the functionality of reconsolidation: B1. Developing mathematical formulation and software models of the Reconsolidation Attractor Network (RAN) that demonstrate the properties and functional benefits of reconsolidation. The RAN provides flexible memory and has the desirable property of having the number of memory attractors independent of the input dimension, thus being free of memory saturation. Furthermore, these

memories can be loaded on-line as in symbolic memories. B2. Developing mathematical formulation and software applications of the Kernel Based Memory (KBM) based on the mathematical theory of kernel functions and the related advances in statistical machine learning, as previously used in support vector machines (by Vapnik) and support vector clustering (by my group and in collaboration with Vapnik). We demonstrate the neural relevance of kernel theory and use it to explain flexibility as seen in existing data on reconsolidation in animals and humans, resulting in an extremely useful engineering tool. This memory is superior in real-time tracking, on-line recognition, and clustering of dynamically moving and changing objects. It works on both continuous and binary inputs, unlike state of the art methods in case based reasoning and in cognitive architectures, which are bound to symbolic information. Another unique property of this memory is that it can store and recall memories of unbounded amount and independent of input dimension, both theoretically and in practical numerical experiment.

C. Concise Accomplishments

We achieved our stated objectives by the design of two new attractor based memory systems. Unlike previous memory networks which load information by being presented static images, frequently with repetitions of the same images, here input comes realistically; images may change with time and the memory can retrieve and update accordingly. This approach put a new spin on the current state of the art in Machine Learning. Our newly designed memories are not bounded a priori by the number of memories, which are independent of input dimension. These memories demonstrate an efficient loading and retrieval algorithms and have the possibility of flexibility after loading. Until now, this combination of features has been considered impossible in the field of computational machine learning. In the Reconsolidation Attractor Network (RAN) attractors can be simply added, deleted, and updated on-line without harming existing memories. The RAN incorporates both fixed and flexible (reconsolidated) memories, a controlled flow with early stopping, and contextual effects. The model shares the properties seen in reconsolidation by proposing particular algorithms that change the attractors during this process. The Kernel Based Memory (KBM) includes the above stated attractive properties as in the RAN, having stronger mathematical support and being more practical in use. The KBM can use both binary and continuous-valued inputs. In terms of neural representation, the KBM is on the one hand a generalization of Radial Basis Function networks and on the other hand it is, in feature space, analogous to a Hopfield network. Input vectors do not have to adhere to a fixed or bounded dimensionality and input may increase and decrease dimensionality without the need to relearn previous memories. This latter property has never been suggested in neural memory models and it is very attractive both for psychological models and for practical applications. It is reminiscent of memories reconsolidated from basic knowledge to full expert knowledge or from memories transferred by emotion and attention to a state of higher importance, and thus containing more details. A continuous version of our network is suggested for modeling firing-rate dynamics. The discrete time version along with its algorithm of reconsolidation enables the network to generalize concepts and form clusters of input data, while input arrives from dynamic, realistic streams with superior results. Our method's efficacy is demonstrated through its ability to recognize head movements, follow a series of morphing faces, and track moving objects, such as missiles.

Using these models, we simulated the order-dependent property seen in reconsolidation in neurophysiology and in psychophysics. We then, considered our model's actual memory representation to observe the actual representations at the beginning, during, and at the end of a process of following a series. We compared such representations to memory that learns from input samples, which originated in a trajectory, but were presented after shuffling. With these, we proposed general principles of reconsolidation-like processes in analog-symbolic memories. The result of our research caused the introduction of these highly efficient methods to the field of Machine Learning.

D. Expanded Accomplishments

Reconsolidation is a storage process distinct from the one time loading employed in consolidation. It serves to maintain, strengthen and modify existing memories shortly after their retrieval.

Being a key process in learning and adaptive knowledge, problems in reconsolidation have been implicated in disorders such as Post Traumatic Stress disorder (PTSD), Obsessive Compulsive disorder (OCD), and even Addiction. Part of the recent growing interest in the reconsolidation process is the hope that controlling it may assist in psychiatric disorders such as PTSD or in the permanent extinction of compulsive fears.
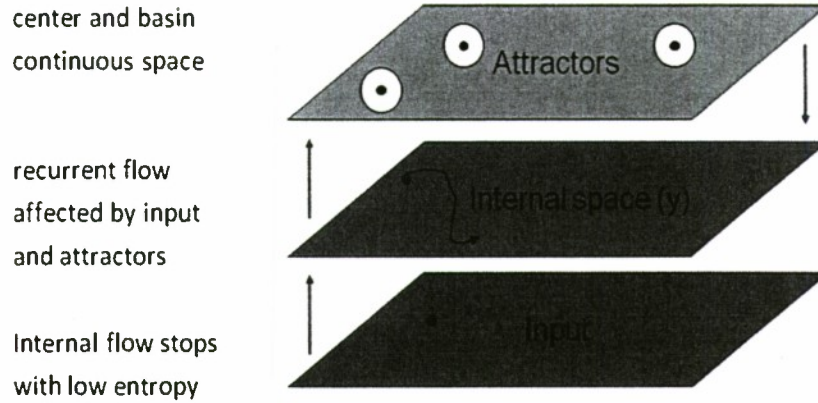
To understand reconsolidation we first analyzed existing studies and modeled them. A property that arises in all reconsolidation demonstrations is that memory representations are sensitive to the order of examples in the input stream. When examples change order, reconsolidation acts effectively to learn and update the gradual changes of objects. When examples are shuffled and the consistent direction of change is lost, existing memories do not update. This property is fundamental in our models. Another conclusion we reached by analyzing existing reconsolidation experiments is that the number of memories in the memory system cannot be a priori bounded and that it must be independent of input dimension. This property is fundamental when thinking in psychological terms, but somehow was not brought up in the main stream of memory modeling. We also suggest, based on mathematical principles, that reconsolidation does not affect only one memory attractor at a time, but rather the neighboring memories must be updated as well. Reconsolidation appears as a continuous phenomenon, yet it occurs in symbolic memory as well, thus the combination of symbols and continuous representations must lie in the brain side by side and inform each other.

Following, we describe our introduction of high-level attractor systems that enable the study of memory reconsolidation properties from both the computational (behavioral) level and the algorithmic (functional) level. This would inform both neurosciences by characterizing the possible mechanisms of flexible memories, as well as computer science and engineering by introducing possible methods for memories that are flexible enough to handle dynamic environments.

D1: Reconsolidation Attractor Network (RAN):

In the RAN model each memory is an attractor, the representation currently believed to underlie the persistent dynamics of memory. This model also fits, so called celebrity neurons, in which particular cells code for abstract concepts that may include different representations. such as a person's image, voice, name, identifying title, etc. Our RAN architecture consists of two levels. The first, which we call the state of the system, is based on state nodes or cells and enables the flow from input to an attractor. Different inputs may have overlap in the associated internal states. The state level of our system is reminiscent of neural network approaches. In the second level, each attractor is represented by a unique node and thus the attractors do not overlap even if the states generated by them would have high overlap.

5

center and basin
continuous space

recurrent flow
affected by input
and attractors

Internal flow stops
with low entropy

$$H_a(t) = \sum_{i=1}^{n} a_i(t) \cdot \log_2(\frac{1}{a_i(t)})$$

Change attractors (c_i) with recall

$$c_i(t+1) = \nu a_i(t) \cdot y(t) + [1 - \nu a_i(t)] \cdot c_i(t)$$

Figure 1: The architecture of a Reconsolidation Attractor Network. The middle layer of internal space is a neural network, and is affected by both inputs and previously learned attractors. Attractors can change when the network gets close to them and flow stops.

As part of our work, we introduced a possible algorithm to merge memories, or more generally to update a memory when the system receives a monotonic sequence of inputs, starting with an input associated with one memory attractor and going all the way to a different one. We demonstrate it by the task of recognizing a person who grows a beard. The memory model of the person growing a beard starts showing a growth of a beard as well, so that if the person arrives one day without a beard he would cause a sizable surprise. We also explain how related memory models show some modifications as well, which in our example translates to not having a big surprise if similar people also appear with a beard. Far memory models will not be affected by the monotonic updates; in particular the system will still be surprised if a woman or a baby appeared with a beard. RAN also demonstrates what happens to an attractor and what it represents after modifications. We ran the same experiment with different entropy values that affect both the measure of surprise and the stopping condition. Higher entropy in the stopping criterion causes bigger changes to near-by attractors because attractor activity distribution is not highly peaked. Additionally, the activity of closer attractors is not significantly different from the activity of the winning one. Lower entropy conditions halt the update of the internal nodes in a more peaked distribution, thus an attractor that does not win, has much lower activity and is affected very slightly by the input.

Contextual effects are demonstrated for the sequence 505 / SOS, see Fig. 2.
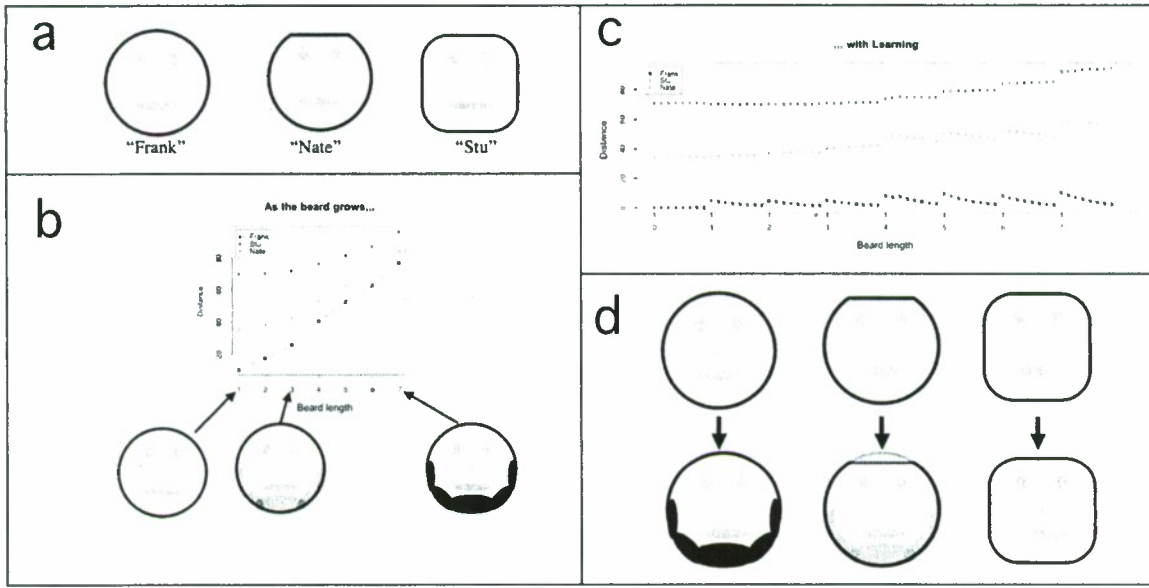
Figure 2: Memory concepts change with a monotonic input sequence that leads toward a new concept: (a) Three faces are stored as non overlapping attractor memories (b) Seven inputs arrive sequentially featuring Frank growing a beard. The distance of each attractor from each input is depicted for when the attractors are held static. The Frank attractor increases its relative distance from bearded Frank (c) The distances of the three attractors from the seven inputs when attractors are flexible (d) The modified attractors are depicted: Frank changes to a bearded Frank, Nate will recognize both clean shaved and bearded Nates, and the Stu memory has not been modified.

Figure 3: Contextual effect due to persistent continuous activity in the state nodes biases interpretations. (a) The high dimensional space of letters and digits is viewed in 2D via PCA applied to the image concatenated with the binary identification column. (b) Input starts with the digit 5 followed by 50% of 0-O and then by 50% of 5-S. The flow after the presentation of the first digit is depicted in red, the flow after the presentation of the 0-O is blue, and the flow after the presentation of the third input is green. The trajectory flows to an unstable middle point 5-S and then biased to 5. (c) When the first input is S, the same sequence leads to final recognition of S. The state nodes leave traces of previously seen inputs, which act as the prior bias to perception for the next input.
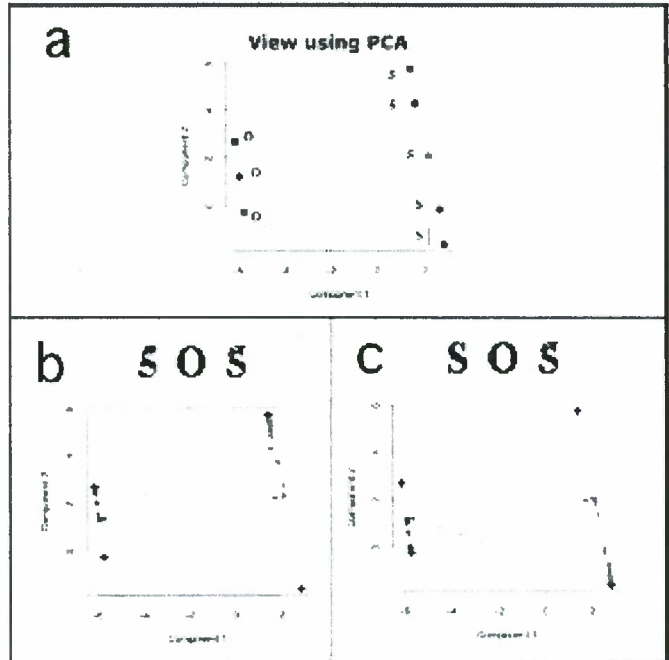
In the global view of the memory system, we have proposed a specific understanding of memory reconsolidation, employing three key elements: compositional generative models, on-line learning, and input-driven dynamic attractors. The first, compositional models, refers to models generated and transferred from long term into short term memory - demonstrating the ability to manipulate existing memories stored as prototypes, e.g., rotating 3D objects. Based on this ability, we propose storing prototypes and operators in long term memories that help to manipulate and associate them. The manipulated prototypes become the instances of the models to reside in short term memory.
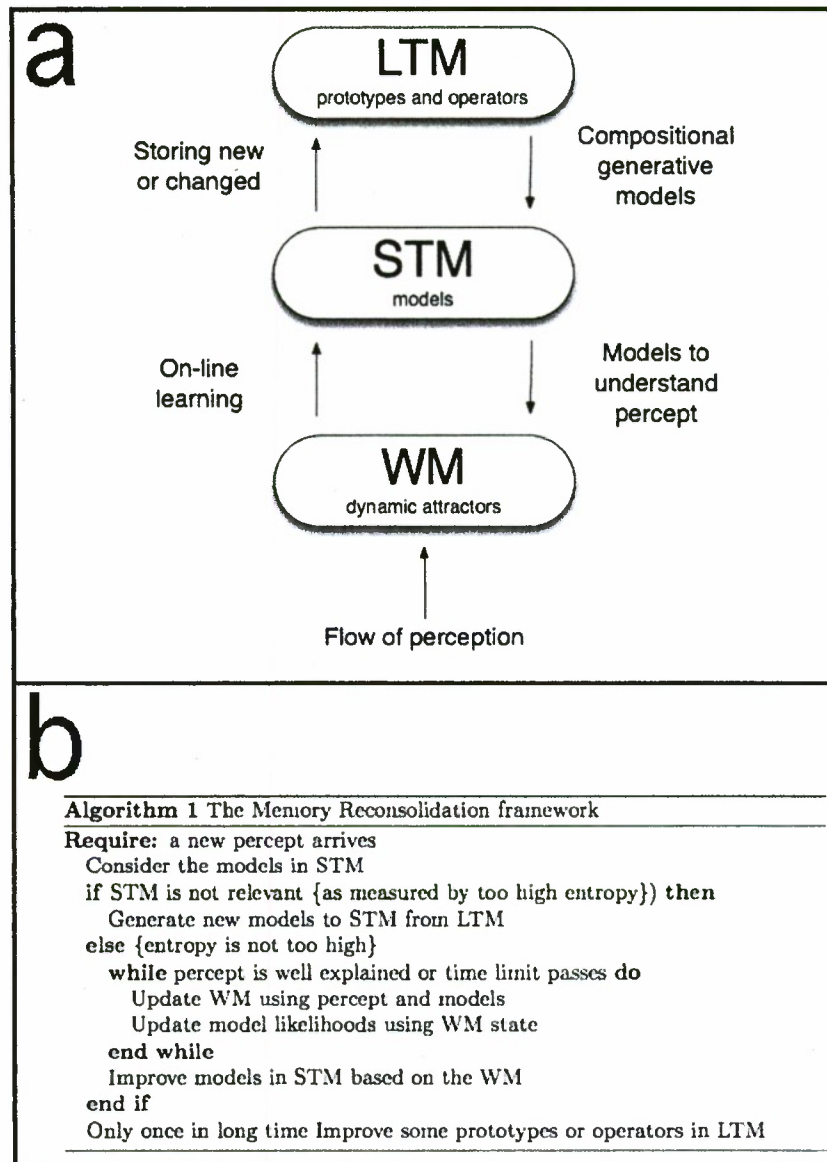

a

LTM
prototypes and operators

Storing new or changed

Compositional generative models

STM
models

On-line learning

Models to understand percept

WM
dynamic attractors

Flow of perception

b

Algorithm 1 The Memory Reconsolidation framework

Require: a new percept arrives
  Consider the models in STM
  if STM is not relevant {as measured by too high entropy}) **then**
    Generate new models to STM from LTM
  else {entropy is not too high}
    while percept is well explained or time limit passes **do**
      Update WM using percept and models
      Update model likelihoods using WM state
    end while
    Improve models in STM based on the WM
  end if
  Only once in long time Improve some prototypes or operators in LTM

Figure 4: The global view of memory reconsolidation with RAN

The second, on-line learning and updates of memories in short term memory, is a necessary element for a system that has to deal with a world that is stochastic and dynamic like ours. The ability to update models over time results in better specializations and generalization, error correction and the tracking of dynamical models as they monotonically change. Short term memory updates should occur during perception and action and not as a separate process, and hence we include the update of memories in our RAN. The third, the study of input-driven dynamic attractors, refers to the attractors rising in state space of the system. Continuous changes to existing attractors occur when a stream of inputs is mixed with different levels of attention and top-down direction. This study provides a dynamical system explanation to the context-sensitivity of memories, see Fig. 4

The RAN model allows us to propose predictions in agreement with mathematical analysis and compare them with biological and psychological data. We suggest that modifications will not occur only to the same memory that has been manipulated by monotonic changes in the input, but also to other related memories. We also propose that the process that causes tracking of dynamic concepts is the same process that causes memory loss and we suggest how to entice or stop this process. We predict that different temporal attention may lead to different perception and different alterations of memories.

Results appeared in H.T. Siegelmann, "Analog-Symbolic Memory that Tracks via Reconsolidation," *Physica D: Nonlinear Phenomena*, 237 (9), 2008: 1207-1214.

D2: Kernel Based Memory (KBM):

KBM is a model whose memory attractors do not lie in the input space, but rather in an implicit feature space with large or infinite dimension, giving rise to an unbounded number and size of memories. This model is isomorphic to the symmetric Hopfield network in the feature space spanned by the kernels, giving rise to a Lyapunov function for the dynamics of associative recalls, enabling the analogy between memories and attractors.

The advantages of this novel approach to attractor memory are many. The input space is naturally composed of either continuous-valued or binary vectors. The number of attractors is independent of the input dimension, thus posing a saturated-free model that does not suffer from corrupted memories with memory overload. The amount of memory can scale up to any desired amount.

In terms of flexibility, attractors are efficiently loaded, deleted, and updated on-line as in the RAN. A very attractive property, which we found and intend to develop further, concerns the fact that input dimensions can change for the different input strings with no a priori bound. This is different from all current associative memory models that require fixed input dimension. This property corresponds to the ability to remember data with more or fewer details and is very relevant for psychological modeling as well as engineering applications where different inputs are represented with different amounts of details.

The process of consolidation in the kernel memory results in attractors in feature space and Voronoi-like space partitions that can be projected efficiently to the input space and describe clusters there, along with their basins of attraction. The process of reconsolidation enables the tracking of monotonic updated inputs, including moving and changing objects. Compared to biological and psychological data, memory representations resulting from reconsolidation were shown to be sensitive to the order of examples. When examples change orderly, the reconsolidation acts effectively to learn and update to the gradual changes of objects. When examples are shuffled and consistent direction of change is lost, existing memories do not update. We show the importance of input ordering in the KBM and how it works in flexible environments and with large-scale data beyond [1]. The advances cited are a significant step toward creating Artificial Intelligence via neural networks at the human level.

Our network can be thought of as generalizing Radial Basis Function (RBF) architectures. Classical RBF networks [2] are 2-layered feed-forward networks, with one RBF and one linear layer. Recurrent versions inherit this 2-layered architecture and add time-delayed feedback from outputs to inputs. Our network enables a more general neural architecture; the neurons can assume a large variety of kernel activation functions and thus distinguish attractors that are similar or highly correlated. Furthermore, the kernel function can be changed during learning to reflect change in input dimension. We further prove that the attractors are either fixed points or 2-cycles, unlike general recurrent RBF networks that may have arbitrary chaotic attractors; regular attractors are advantageous for memory systems.

The memory system introduced here takes advantage of kernel methods and the theory introduced in the Support Vector Machine (SVM) [3], the leading classifier in the field of machine learning, and in Support Vector Clustering [4]. In support vector clustering, clusters are formed when a sphere in the φ-space spanned by the kernels is projected to input space. Here the clustering is a side effect of the consolidation process that creates memories as separated fixed points in the φ-space, and where the Voronoi polyhedron is projected on the formation of clusters in the input space. On top of it, clustering can be made dynamic during changes of inputs, improving the current state of the art in clustering.

D.2.1 Kernel Hetero-associative and Auto-associative Memories

A general framework of heteroassociative memory is defined from input to output space. The input vectors can be written as the columns of matrix X (n ×m) and the associated vectors in the output space as the columns of matrix Y (p×m). A projective operator, i.e., a matrix transfers from X to Y. In order to overcome the common dependence of memory capacity on input dimension, we transform the input space to a new input space, which we call feature space, whose dimensionality is greater than n (it could even be an infinite-dimensional Hilbert space). The transformation φ is considered to be transferring from input to feature space. The kernel associative memory algorithm is written as follows:

(1)

(2)

10

with ``+" being the Moore-Penrose pseudoinverse. If the columns are linearly independent, the pseudoinverse can be calculated by

$$[\varphi(X)]^+ = ([\varphi(X)]^T \varphi(X))^{-1}[\varphi(X)]^T \tag{3}$$

Defining S as

$$S = [\varphi(X)]^T \varphi(X)$$
$$s_{ij} = (\varphi(x_i), \varphi(x_j)), \tag{4}$$

the memory loads by:

$$\mathbf{B} = \mathbf{Y}\mathbf{S}^{-1}[\varphi(\mathbf{X})]^T. \tag{5}$$

and the recall procedure is calculated by:

$$\tag{6}$$

We note that during both loading (5) and recall (6) procedures, the function φ appears in the pair $(\varphi(w), \varphi(v))$. We can thus define a Kernel function and gain computational advantage. We write S and z using the Kernel K:

$$\tag{7}$$

The Kernel function is a scalar, and thus even if φ was a function of high dimension the calculation of the multiplication is a scalar and thus efficient.

This memory is proven to associate loaded pairs correctly and to associate close by values otherwise. Furthermore, the kernel heteroassociative memory has no a priori bound on capacity in the following sense: for any given number of memories m we can find a kernel K such that the memory with this kernel will provide the correct association.

Autoassociative memory: We next focus on the special case where input is associated to itself. Here the loading algorithm is the same as above and recall is facilitated by the iterative form

$$\tag{8}$$

The activation function f, applied by coordinates, is a generalized sigmoid: it needs only to be a bounded monotonically increasing real-valued function over R such that its left limit approaches a, its right limit approaches b, and $b>a$. We prove that the recall procedure always converges and that the attractors are either fixed points or 2-limit cycles. See Fig. 5.

11
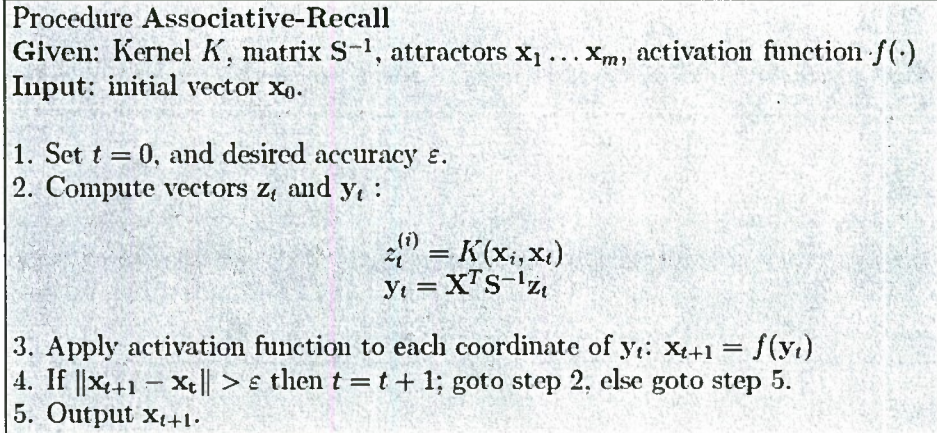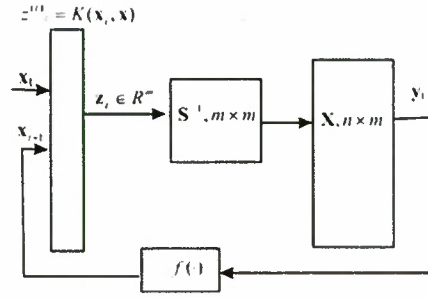
$$z^{(i)}_t = K(x_i, x)$$

Procedure Associative-Recall
Given: Kernel K, matrix S^{-1}, attractors $x_1 \ldots x_m$, activation function $f(\cdot)$
Input: initial vector x_0.

1. Set $t = 0$, and desired accuracy ε.
2. Compute vectors z_t and y_t :

$$z^{(i)}_t = K(x_i, x_t)$$
$$y_t = X^T S^{-1} z_t$$

3. Apply activation function to each coordinate of y_t: $x_{t+1} = f(y_t)$
4. If $\|x_{t+1} - x_t\| > \varepsilon$ then $t = t + 1$; goto step 2, else goto step 5.
5. Output x_{t+1}.

Figure 5: Associative recall: flow chart and algorithm

D.2.2 Neural Networks Representation

The autoassociative kernel memory can be directly implemented in a recurrent layered neural network (Fig. 6a): The network has n inputs. The first layer has m neurons that perform kernel calculations; the i-th neuron computes:

The second layer has m neurons with weight matrix S^{-1}. The neurons of the second layer can be either linear or have the generalized sigmoid activation function. The third layer also has n neurons and its weight matrix is X^T. Its activation function can be linear or generalized sigmoid. The network has "one-to-one" feedback connections from the last layer to the inputs. In recall mode it works in discrete time, like Hopfield networks.

Maximizing Neural Capacity: We can maximize the network capacity by approximating S by the matrix I. This approximation is suitable if the stored patterns are sufficiently distant in the kernel view. With this approximation one can save m^3 connections without significant loss of association quality by eliminating the middle layer in Fig 6a and the other two layers will have weight matrices X and its transpose; see Fig. 6b. So, to store m vectors of dimension n we would need mn real numbers only (lossless coding). The memory capacity connections/neurons ratio is now larger than 1.

Robustness: A key question for any neural network or learning machine is how robust it is in the presence of noise. We prove that there is a pretty large attraction radius where patterns within go to the right memories.
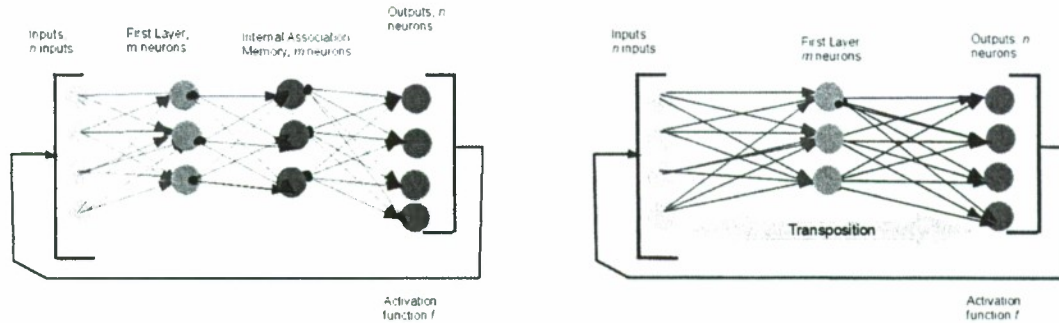

Figure 6: (a) A neural-network that directly corresponds to the algorithm of learning and recall as described in a previous section. (b) Using approximation, the network can be minimized and the capacity maximized.

D.2.3 Flexibility in the Attractor and Input Spaces

The kernel associative memory can be made capable of adding and removing attractors explicitly. To add a new attractor to the network we create a new neuron in the S matrix layer. The dimension of the matrix S is increased from m to $m+1$ and we update the inverse of S efficiently using the linear-algebra identity:

$$(A + B)^{-1} = A^{-1} - A^{-1}B(I + A^{-1}B)A^{-1} \tag{9}$$

Similarly one can delete an attractor by reducing the dimension of S.

We also propose a mechanism that enables the network to handle heterogeneity of input dimension with no need to relearn the previously learnt inputs. Assume the current dimension in the input space consists of the ``initial dimension" $n$ and the ``new dimension" q. We will choose a new kernel that combines the dimension. To save operations we will focus on kernels that can be written in an additive form:

$$K_{n+q}(x,y) = K_n(x_a, y_a) + K_q(x_b, y_b) \\ + K_{int}(x_a, y_b) + K_{int}^*(x_b, y_a) \tag{10}$$

We prove that a small alteration to the kernels enables changing input dimensionality without losing previously learnt attractors.

13

D.2.4. Memory Consolidation and Reconsolidation

The memory system with its loading algorithm enables consolidation of inputs into clusters using the competitive learning method where only the closest attractor is being updated. For reconsolidation we chose a global update, while retaining the property, in which the closest attractor is updated most.

A model of reconsolidation, based on Hebbian learning, was introduced in [1]. The update was based on additions and scalar multiplications in matrix operations. In our kernel associative memory, the corresponding space is no longer linear but rather is a Riemannian manifold. Additions and multiplications by a scalar are not defined in this space. To remedy the situation we define a Riemannian distance and a geodesic, which enables the memory to change gradually as new but close stimuli arrive. Suppose that initially we have a memory $X(0)$ that contains m attractors. Then we obtain $X(1)$ by replacing one attractor by a new stimulus that flows to it. The distance between $X(0)$ and $X(1)$ can be thought of as a measure of ``surprise'' at the memory experience when it meets new stimuli. To reconsolidate, the memory moves slightly on the manifold from $X(0)$ to $X(1)$. See the algorithm in Fig. 7.

Procedure **Geodesic-Update**

Given: Current state of the memory $\mathbf{X}_t^0$, parameter $\alpha \in [0; 1]$

Input: Current stimulus $\mathbf{x}_t$, time t.

1. Run **Associative-Recall** of a memory $\mathbf{X}_t^0$ on input $\mathbf{x}_t$, denote the recall by $\mathbf{x}_t'$

2. **Remove-Attractor** $\mathbf{x}_t'$ from $\mathbf{X}_t^0$

3. **Add-Attractor** $\mathbf{x}_t$ to the memory. Denote the result as $\mathbf{X}_t^1$

4. Build a geodesic γ between $\mathbf{X}_t^0$ and $\mathbf{X}_t^1$. Denote its length by L.

5. Take a point $\mathbf{X}_t^*$ on γ in distance αL from $\mathbf{X}_t^0$.

6. Set time $t + 1$ Output $\mathbf{X}^{t+1} = \mathbf{X}_t^*$.

Figure 7: The reconsolidation algorithm.

A few demonstrations are shown next.

Static consolidation: The algorithm was applied on the MNIST database of handwritten digits [5]. A Gaussian kernel was chosen

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(\frac{-\alpha}{2R^2} \sum_{k=1}^{d} w_k (x_k - y_k)^2 \right)$$

with

When only the best attractor is tuned, which we consider consolidation, the best recognition was 91.1%, see Table 1. Our classification is slightly superior to other unsupervised clustering techniques.
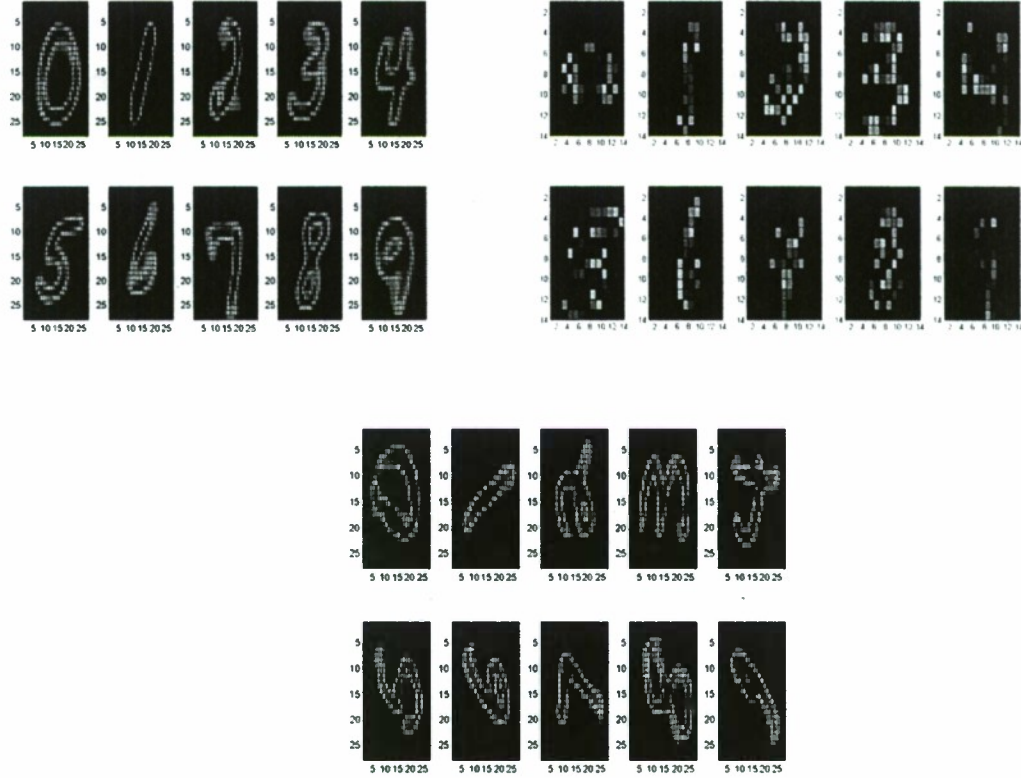

Figure 8: Top: original versus downscaled images. Bottom: rotated digits.

Clustering during change of input dimensionality: Training set was divided into two. The first half was given with a resolution of 14 ×14 (Fig. 8, top right) and the second half with full image size (28 ×28). The recognition quality went from 78.12% to 91.66% when each set was of size 10,000 and it was slightly worse when each set was of 5,000 examples, see Table 1.

Reconsolidation algorithm with rotating digits. A learning set of rotating digits was created. It contained 90,000 images obtained from 1,000 original digits (100 per class) by rotating them counterclockwise on angles from zero to 180 deg. In one experiment we first clustered the static images and then reconsolidated on the rotating images, we obtained 94.18 recognition rate. In a following experiment we relied only on the rotating input stream without prior classification: attractors were initialized with random digits from the whole database. We stopped looking at the input when reaching same excellent results, see Fig. 8 bottom, and Table 1.

15

Operation	#Train Images	#Test Images	Class. Rate, %	Note
Consolidation	10,000	10,000	91.38	
Reconsolidation on static dataset	10,000	10,000	90.72 to 91.51	Depends on sample ordering
Consolidation on downscaled, 14*14	10,000	10,000	78.12	
Add full size 28*28 images	+10,000	10,000	91.66	Tested on both small and large images
Like above but with total 10,000 inputs	5,000 small +5,000 full	5,000 small +5,000 full	76.8 then 89.2	
Classifying the static, and then re-consolidate the stream	10,000 straight +90,000 rotated	10,000	94.18	Tested on 10,000 images closest to final state
Reconsolidation on rotated digits, no prior consolidation	95340–96780 (Depends on initialization)	10000	>94.18	Test as in the above row, stop the input stream when reaching the desired threshold of 94.18.

Table 1: Clusters with kernel memories are superior to previous tests on the MNIST database. In all simulations the network had 1000 attractors, 100 attractors per class.

Morphed faces. The goal of this experiment is both to show the performance of the reconsolidation process we describe on large-scale data and to compare its properties with the recent psychological study in [6]. We used the database Productive Aging Lab Face Database [7]. Faces were morphed using the software Sqirlz morph. Original size of all images was 640×480. The useful area fell in the rectangle 320 × 240, and images were cropped to this size before being entering into the network. The database contained 150 morph sequences, each of them consisted of 100 images.

In our simulations we created a network with 16 attractors representing 16 different faces; it had 76800 input and output neurons, and two middle layers of 16 neurons each. Four arbitrarily selected network attractors are depicted in Fig. 9. A Gaussian kernel was chosen in order to simplify calculations with large scale data.

When the learning order followed image order in the morphing sequence, attractors changed gradually and consistently. The ability to recognize the initial set of images gradually decreased when attractors tended to the final set. In the case of random learning order, attractors quickly became senseless, and the network was not able to distinguish faces. This experiment also demonstrates the efficiency of the reconsolidation processing kernel memories for high-dimensional data.
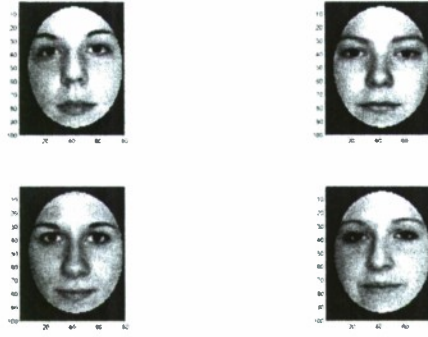
Figure 9: Morphed faces, examples of attractors during reconsolidation

Rotating heads. This example focuses on rotating head images for reconsolidation based on the VidTIMIT dataset[8]. A video of each person is stored as a numbered sequence of JPEG images with a resolution of 512 by 384 pixels. The ability to track and recognize faces was tested on a set of 15 last frames from each sequence. With reconsolidation and ordered stimuli, the obtained recognition rate was 95.2%. If inputs were shuffled randomly, attractors got messy after 30-50 updates, and the network did not demonstrate significant recognition ability. It can be seen how attractor images are blurred when head movement is fast (Fig. 10).
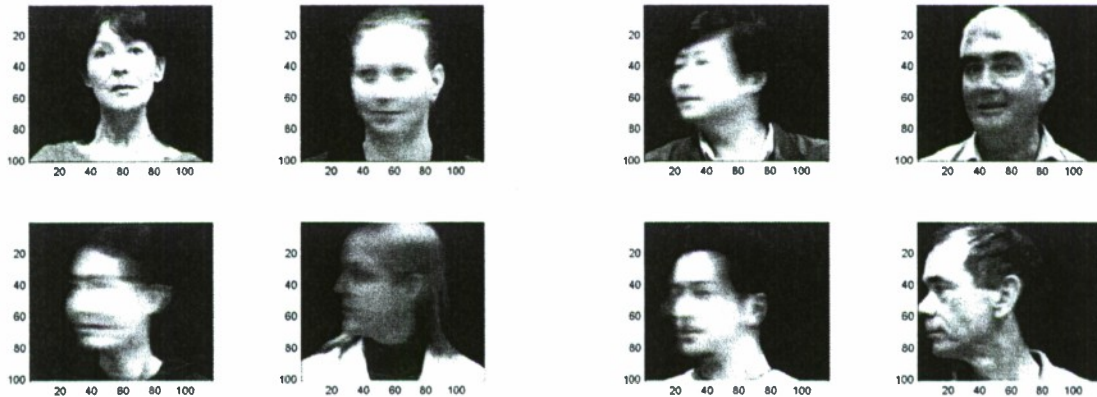

Figure 10: Tracking rotating heads via reconsolidation. Attractors are blurred in fast motion

Tracking the Moving Patriot: We analyzed videos of Patriot missile launches with resolution 320 by 240, originally in RGB color, and transformed them to grayscale. The memory was loaded with a vector composed of two 40 by 40 pixel regions (windows) around the missile taken from two consequent frames and a two-dimensional shift vector indicating how the missile center has moved between these frames. Optimal number of attractors was found to be 16-20. Using memory reconsolidation algorithm we were able to calculate velocity vector every time, and therefore track the missile with great precision, with only average error of 5.2 pixels, see Figure 11.
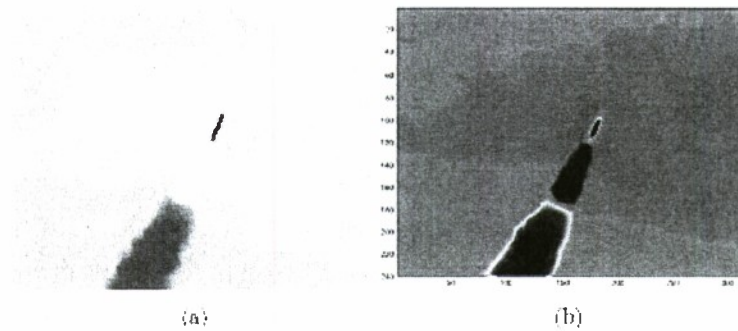
Figure 11: Patriot missile. Original frame (a) and a processed frame (b).

D.2.5 Summary

We proposed the design of memory systems that can be used for improved thinking machines that are able to follow dynamically changing concepts and demonstrate sensitivity to context. The same memories are also useful in testing hypotheses regarding reconsolidation-like processes and dynamic memory tuning, which are relevant to human flexible memories. The new computational machines naturally combine learning from examples, high-level directions, and cognitive-like attention, and thus may change the state of the art of machine learning which is currently best equipped to produce rigidly single task oriented algorithms and handle and cluster static data.

References

[1] B. Blumenfeld S. Preminger D. Sagi and M. Tsodyks. Dynamics of memory representations in networks with novelty-facilitated synaptic plasticity. Neuron, 52:383--394, 2006.

[2] T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. Science, 247:978--982, 1990.

[3] V. Vapnik. Statistical Learning Theory. John Wiley & Sons, NY, 1998.

[4] A. Ben-Hur D. Horn H.T. Siegelmann and V. Vapnik. Support vector clustering. Journal of Machine Learning Research, 2:125-137, 2001

[5] Y. LeCun L. Bottou Y. Bengio and P. Haffner. Gradient-based learning applied to document recognition. Proc. IEEE., 86(11):2278-2324, 1998.

[6] S. Preminger D. Sagi and M. Tsodyks. Morphing visual memories through gradual associations. Perception Suppl., 34:14, 2005.

[7] M. Minear and D.C. Park. A lifespan database of adult facial stimuli. Behavior Research Methods, Instruments, & Computers., 36:630-633, 2004.

[8] C. Sanderson. Biometric Person Recognition: Face, Speech and Fusion. VDM-Verlag, 2008.

E. Productivity

I) Refereed journals:

H.T. Siegelmann, "Analog-Symbolic Memory that Tracks via Reconsolidation," *Physica D: Nonlinear Phenomena*, 237 (9), 2008: 1207-1214.

A. Z. Pietrzykowski, R. M. Friesen, G. E. Martin, S.I. Puig, C. L. Nowak, P. M. Wynne, H. T. Siegelmann, S. N. Treistman, "Post-transcriptional regulation of BK channel splice variant stability by miR-9 underlies neuroadaptation to alcohol," *Neuron,* 59, July 2008: 274-287.

K. Tu, D. Cooper and H.T. Siegelmann, "Memory Reconsolidation for Natural Language Processing," *Cognitive Neurodynamics*, 3, 2009: 365-372.

M.M. Olsen, N. Siegelmann-Danieli and H.T. Siegelmann, "Robust Artificial Life Via Artificial Programmed Death," *Artificial Inteligence* 172(6-7), April 2008: 884-898.

F. Roth, H. Siegelmann, R. J. Douglas, "The Self-Construction and -Repair of a Foraging Organism by Explicitly Specified Development from a Single Cell," *Artificial Life* 13(4), 2007: 347-368.

W. Bush and H.T. Siegelmann, "Circadian Syncrhonicity in Networks of Proteim Rhythm Driven Neurons," *Complexity* 12(1), September/October 2006: 67-72.

D. Nowicki and H.T. Siegelmann, "Flexible Kernel Memory," *PLOS One,* under changes.

H.T. Siegelmann and L. Holtzman, "The brain can perform exact Bayesian inference with learning," submitted.

K. Tu, M.M. Olsen, and H. T. Siegelmann, "Inference and Recognition from Annotated Videos in Security Systems," submitted.

II) Conference proceedings and posters:

K. Tu and H.T. Siegelmann, "Text-based Reasoning with Symbolic Memory Model," Proceedings of the Fifth International Workshop on Neural-Symbolic Learning and Reasoning (NeSy'09), July 2009: 16-21.

D. Nowicki and H.T. Siegelmann, "The Secret Life of Kernels: Reconsolidation in Flexible memories," Computational and Systems Neuroscience (COSYNE), February 2009.

M. Olsen, K. Harrington and H.T. Siegelmann, "Emotions for Strategic Real-Time Systems," AAAI Spring Symposium on Emotion, Personality and Social Behavior, March 2008.

D. G. Cooper, D. Katz and H.T. Siegelmann, "Emotional Robotics: Tug of War," AAAI Spring Symposium on Emotion, Personality and Social Behavior, March 2008.

L. Holtzman and H.T. Siegelmann, "Input driven dynamic attractors," Computational and Systems Neuroscience (COSYNE), February 2007: 101.

L. Holtzman and H.T. Siegelmann, "Exact neural Bayesian learning and inference over graphical models," Computational and Systems Neuroscience (COSYNE), February 2007: III-70.

M. Olsen, H. Siegelmann, "Multi-Agent System that Attains Longevity via Death," Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI), Jan 2007: 1428-1433.

III) Book chapter:

K. I. Harrington and H. T. Siegelmann, "Adaptive Multi-modal Sensors," in *50 years of Artificial Intelligence*, Lungarella, F. Iida, J. Bongard, R. Pfeifer (eds.) Springer 2007: 264-173.

III) Talks:

After Dinner Talk: The 8th Understanding Complex Systems Symposium, Urbana- Champaign, May 2008.

Plenary talk: International Conference on Complex Systems (ICCS), Boston, October 2007.

Plenary Talk: Unconventional Computation, Santa Fe, March 2007.

Plenary Talk: Understanding Complex Systems, Urbana Champaign, May 2007.

Invited talk: University of Zurich Laboratory of Artificial intelligence Summer Course in Biological Inspired Computation, Switzerland, June 2006.

Invited talk: Frankfurt Institute for Advanced Studies (FIAS) Department of Physics, Germany, July 2007.

F. Technology Transfer

Our study provides a functional algorithmic understand to flexible memory tuning as occurring during retrieval, that is able to follow dynamically changing concepts and cluster them on-line with sensitivity to context. Importantly, the number of memories is independent of the input dimension and thus can grow as needed. It is possible that this new approach to memory can be embedded in robots to provide both a machine with optimal tracking capabilities as well as one that interacts smoothly with humans.